U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEIC 2º Ano

Janeiro 2019

ALGORITMOS E ESTRUTURAS DE DADOS

Trabalho Prático – Parte 2

Tema 1 - Transporte Escolar

Turma 5 – Grupo 7:

João Praça - up201704748@fe.up.pt

Leonor Sousa - up201705377@fe.up.pt

Sílvia Rocha - up201704684@fe.up.pt

Prof.ª Ana Paula Cunha da Rocha

Prof. Luís Paulo Gonçalves dos Reis

Índice

Descrição do Tema

O tema abordado neste projeto tem por objetivo implementar um sistema de gestão adequado e eficiente para empresas de transporte escolar.

Para além de todas as características descritas e implementadas na primeira parte do projeto, o sistema inclui três funcionalidades extra.

A primeira corresponde a sistema de armazenamento das várias escolas associadas à empresa, através da utilização de uma árvore binária. Esta estrutura de dados permite, facilmente, aceder ou procurar uma escola de acordo com o número de utentes associado e o seu nome. Tal funcionalidade revela-se útil quando é necessário obter a informação de qual a escola que mais trabalho fornece à empresa e/ou quais as escolas que podem estar a dar prejuízo à empresa (visto terem um diminuto número de utentes associados).

A segunda funcionalidade consiste num sistema responsável pelas reparações dos vários veículos. Cada veículo precisa de reparações (programadas ou não), devendo esta reparação ser efetuada pela oficina que esteja disponível mais cedo e que não esteja a uma distância demasiado grande da garagem da empresa onde são guardados todos os veículos da mesma. Cada oficina demora um dia para reparar um veículo.

A terceira e última funcionalidade caracteriza-se por um sistema de gestão dos vários motoristas da empresa. O sistema deve, então, guardar todos os motoristas que trabalham ou já trabalharam para a empresa. Quando for necessário à empresa contratar um motorista, os motoristas antigos deverão ter sempre a preferência em relação a um motorista desconhecido à empresa, daí a necessidade de guardar motoristas antigos.

Descrição da Solução Implementada

Para o problema acima descrito, desenvolvemos uma solução que combina a manutenção da estrutura do sistema implementado na 1ª parte do projeto com a eficiência desejada neste tipo de sistema. Desta forma, foram efetuadas as seguintes modificações e ampliações às classes já implementadas:

- Modificação da classe *Escola*, de forma a ter três mais atributos: o número de utentes, a morada da escola e o nome do seu diretor;
- Criação da classe *EscolaPtr*, que tem um único atributo, um apontador para uma *Escola*;
- Criação da classe *Oficina*, contendo quatro atributos: uma *string* com o nome da oficina, um inteiro com a disponibilidade da oficina (número de carros que faltam reparar), um inteiro com a distância da oficina à garagem da empresa, e também uma fila de apontadores para objetos da classe *Veículo*, contendo os veículos que estão na oficina para reparo. Foram ainda adicionados métodos para obter estes atributos, alterar a fila de veículos e a disponibilidade (*adicionaVeiculo* e *removeVeiculo*), verificar a existência de um certo veículo na oficina, e o operador <, para efeitos de ordenação correta na fila de prioridade de oficinas presente na classe Empresa.
- Criação da classe *Motorista*, que contém quatro atributos: uma *string* com o nome do motorista, um vetor de datas (classe *Date*) de todos os serviços que lhe foram atribuídos e dois booleanos: um indicativo de se o motorista em questão é trabalhador atual da empresa ou antigo(*isActive*) e o outro que indica se o motorista está associado a algum serviço regular ou disponível para que lhe possam ser atribuídos serviços. Nesta classe, foram também implementados os métodos que retornam cada um destes quatro atributos assim como métodos para alterar os três últimos.
- Modificação da classe *Veículo* e suas classes derivadas. Deste modo, cada veículo adquire dois novos atributos: uma *string motorista* e um *bool underRepair*. Nos veículos de serviço regular, o atributo motorista corresponde ao nome do motorista que efetua, habitualmente, aquele trajeto. Já nos veículos de serviço especial, este atributo representa o motorista que realizou o serviço mais recente com aquele veículo. Em ambos os veículos, *underRepair* é um atributo que permite identificar se o veículo está ou não em reparação. Foram também implementados os métodos que permitem retornar e alterar estes novos atributos.
- Modificação da classe *Empresa*, incluindo como atributos uma árvore binária de pesquisa *escolas*, constituída por elementos da classe *EscolaPtr*, uma fila de prioridades *oficinas*, onde estão contidas todas as oficinas associadas à empresa, ordenadas por disponibilidade (em caso de empate, por maior distância à garagem da empresa, e caso este valor também seja equivalente, por ordem alfabética), e ainda uma tabela de dispersão *HashTabMotorista* que contém todos os atuais e antigos motoristas da empresa. Relativamente aos métodos desta classe, foram adicionadas diversas funções associadas a cada uma destas novas estruturas:
    - ✓ Relacionada com a implementação da árvore binária foi adicionada uma função responsável por listar todas as escolas associadas à empresa, por ordem de número de utentes e, em caso de empate, por ordem alfabética do nome da escola.
    - ✓ Já em relação à fila de prioridade, foram incluídas funções para adicionar, remover, e verificar a existência de uma oficina, adicionar um veículo para reparação numa

oficina específica ou na oficina mais apropriada, reparar um certo número de veículos em todas as oficinas, e por fim a listagem das mesmas.

- ✓ Finalmente, relativamente à tabela de dispersão, foram adicionadas funções para a listagem total ou parcial (ativos e inativos) dos motoristas da empresa e para adicionar ou remover um motorista (verificando se esse motorista já existe).

- Alteração do operador << nas classes *ServicoRegular*, *ServicoEspecial* e *Empresa* assim como no construtor da classe *Empresa* a partir de um ficheiro para que as novas informações sejam incluídas.

Do ponto de vista do utilizador, e como já tinha acontecido na 1ª parte do projeto, existe uma abstração total das estruturas implementadas, interagindo o utilizador apenas com uma interface simples criada para o efeito.

Lista de Casos de Utilização

Para além dos vários casos já ilustrados no relatório da 1ª parte do projeto, aplicam-se muitos outros:

1. **Listagem de dados relevantes da empresa**

Na 1ª parte do projeto foram desenvolvidas três listagens: listagens dos utentes, dos ganhos e da empresa. Nesta 2ª parte, foram implementadas listagens de outros dados relevantes da empresa.

A listagem das escolas associadas à empresa é uma delas. Esta listagem é efetuada por ordem de maior número de utentes da escola associados à empresa e, no caso de o número de utentes ser igual, por ordem alfabética do nome da escola.

Uma outra listagem adicionada foi a listagem total e parcial dos motoristas associados à empresa. Deste modo, é permitido visualizar todos os motoristas associados à empresa com a indicação de se são trabalhadores atuais ou antigos da empresa ou listar parcialmente estes motoristas visualizando apenas os motoristas ativos ou apenas os motoristas inativos.

Por fim, foi incluída a listagem de oficinas, dispostas por número de disponibilidade, e onde é possível visualizar o nome da oficina e o número de veículos que nela se encontram para serem reparados.

2. **Gestão dos motoristas da empresa**

Com a criação da classe *Motorista*, foram criadas também as funções necessárias para que seja possível a adição e remoção (caso haja algum engano na inserção ou por algum motivo a empresa não queira guardar os registos daquele motorista) de motoristas.

3. **Gestão da reparação de veículos da empresa**

Quando um veículo da empresa necessita de reparação, é utilizada a funcionalidade de adicionar o veículo à oficina mais apropriada. As oficinas estão ordenadas por disponibilidade (em caso de empate, por maior distância à garagem da empresa, e caso este valor também seja equivalente, por ordem alfabética), possibilitando que a escolha da oficina seja a que demore menos tempo a repor o carro, num limite máximo de distância fornecido pelo utilizador. Para concretizar a reparação de veículos, como cada oficina repara um veículo por dia, é simulada a passagem do tempo, e, por isso, um certo número de veículos é reparado por oficina.

Aspetos sobre a Realização do Trabalho

Esta segunda parte do projeto teve um processo de trabalho em equipa um pouco diferente da 1ª parte, havendo uma maior divisão de trabalho, devido ao pouco tempo disponível para a sua elaboração.

Inicialmente, foi realizada uma reunião, onde se fez uma divisão de tarefas igualitária: cada elemento ficou responsável pela implementação de uma das estruturas de dados recomendada. A implementação da árvore binária foi a primeira a ser efetuada, visto que era a que implicava maiores alterações no código proveniente da 1ª parte. Após esta implementação, o código foi fornecido aos restantes elementos do grupo para que pudessem implementar a sua parte com base nas alterações efetuadas.

O relatório foi desenvolvido pelos vários elementos do grupo, cada um elaborando a parte que lhe correspondia. No entanto, todos os membros colaboraram para o desenvolvimento de todo o relatório, visto que após uma versão inicial do mesmo, todos deram sugestões para o seu melhoramento.

Desta forma, concluímos que todos os membros de igual modo.